CS 134: Networks

Prof. Yaron Singer | Spring 2017

1 PageRank

ALG 1: PageRank

input: Graph G with M nodes, number of steps t

1. Assign all nodes the same initial PageRank value, $1/n$;
2. For t iterations:

    Update all PageRank values according to the *Basic PageRank Update Rule*;

  End for

return: The PageRank value of each node

Definition. ***Basic PageRank Update Rule:*** *Each page divides its current PageRank equally across its outgoing links and passes these equal shares to the pages it points to. (If a page has no outgoing links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives.*

We can also think about PageRank in terms of transition matrices, which allow us to analyze how the PageRank values of a given graph might converge as the number of updates becomes large. Let M be an n by n matrix, where n is the number of nodes in the network and M_{ij} is the share of page i's PageRank that j should get in one update step. Thus, we have:

$$M_{ij} = \begin{cases} 1/d(i) & \text{if } i \text{ points to } j \\ 1 & \text{if } d(i) = 0 \text{ and } i = j \\ 0 & \text{otherwise} \end{cases}$$

where $d(i)$ is the outgoing degree of i. If we represent the PageRanks of all nodes at step t using a vector $r^{(t)}$, then to calculate an entry in $r^{(t)}$ we have:

$$r_j^{(t)} = M_{1j} \cdot r_1^{(t)} + \cdots + M_{nj} \cdot r_n^{(t)}.$$

This corresponds to multiplication by the transpose of our matrix M so we have that:

$$r^{(t)} = M^T \cdot r^{(t-1)}$$

In terms of our initial PageRank values at $t = 0$, then, we get:

$$r^{(t)} = (M^T)^t \begin{bmatrix} 1/n \\ \cdots \\ 1/n \end{bmatrix}.$$

This all looks a lot like what we did with the voter model and random walks last week. Intuitively, we realize that if we start at a random node in the graph, the probability of being at a page X

after k steps of a random walk using our transition matrix M is precisely the PageRank of X after k applications of the Basic PageRank Update Rule!

Definition. ***Scaled PageRank Update Rule***: *First apply the Basic PageRank Update Rule. Then scale down all PageRank values by a factor of s. We divide the residual $1-s$ units of PageRank equally over all nodes, giving $\frac{1-s}{n}$ to each.*

In layman's terms, we can think of the Scaled PageRank Update Rule as introducing a "teleportation factor" to our algorithm, such that some small fraction of the PageRank of each page will "jump" from to every other page, even those that aren't connected by an edge. This helps us avoid the problem of a "sink" absorbing all of the PageRank in a network! If we are applying the Scaled PageRank Update Rule to our transition matrix M, we calculate new values in our new "scaled" matrix $\hat{M}$ as follows:

$$\hat{M}_{ij} = sM_{ij} + \frac{1-s}{n}$$

Then the following equations all hold true:

$$r_j^{(t)} = \hat{M}_{1j} \cdot r_1^{(t)} + \cdots + \hat{M}_{nj} \cdot r_n^{(t)}.$$

$$r^{(t)} = \hat{M}^T \cdot r^{(t-1)}$$

$$r^{(t)} = (\hat{M}^T)^t \begin{bmatrix} 1/n \\ \cdots \\ 1/n \end{bmatrix}.$$

In a similar vein to our earlier analysis, if we start at a random node in the graph, the probability of being at a page X after k steps of a random walk using our transition matrix $\hat{M}$ is precisely the PageRank of X after k applications of the Scaled PageRank Update Rule.

Definition. *(Review) For a square matrix M, a scalar λ is an **eigenvalue** if there exists a nontrivial solution x such that $Ax = \lambda x$. x is called the **eigenvector** corresponding to λ.*

Exercises: What are the strongly connected components for the network in Figure 1? What is the transition matrix under the Basic PageRank Update Rule? What is the transition matrix under the Scaled PageRank Update Rule with $s = 0.9$?

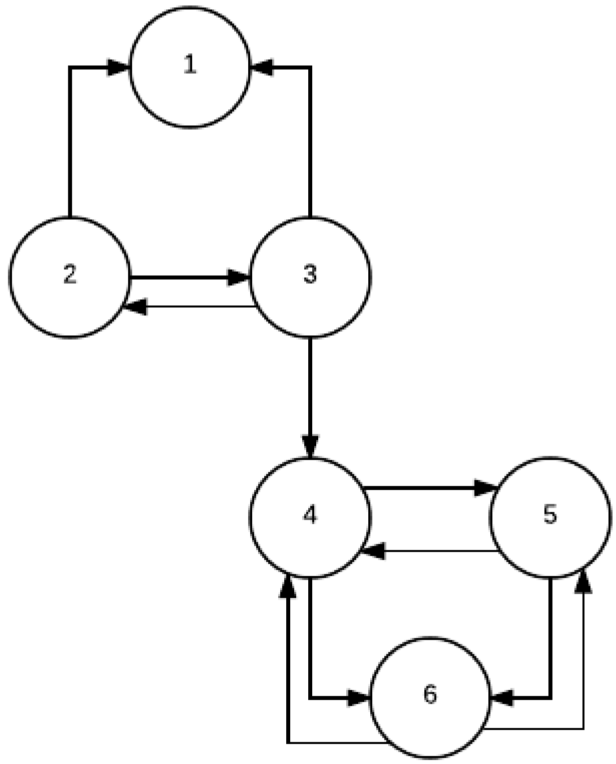

Figure 1

2 Hubs and Authorities

Definition. ***Authority Update Rule***: *For each page p, update a_p to be the sum of the hub scores of all pages that point to it.*

Definition. ***Hub Update Rule***: *For each page p, update h_p to be the sum of the authority scores of all pages that it points to.*

ALG 2: HITS Algorithm

input: Graph G with n nodes, number of steps t

1. Start with all hub scores and all authority scores of each page equal to 1.
2. For t iterations:
   First apply the Authority Update Rule to the current set of scores;
   Then apply the Hub Update Rule to the resulting set of scores;
   End for
3. Normalize by dividing each authority score by the sum of all authority scores.
4. Normalize by dividing each hub score by the sum of all hub scores.

return: The hub and authority scores of each page

Exercise: Perform the HITS algorithm on the network in Figure 1 with $t = 2$ (Feel free to construct an adjacency matrix!).

Thinking about hubs and authorities in a network, we define an **adjacency matrix** M as follows:

$$M_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Then, letting $h^{(t)}$ and $a^{(t)}$ be the vectors of all hub and authority scores respectively at time step t, following the HITS algorithm we have:

$$a^{(t)} = M^T h^{(t-1)}$$

and

$$h^{(t)} = M a^{(t)}$$

More generally, we infer:

$$a^{(t)} = (M^T M)^{t-1} M^T h^{(0)}$$
$$h^{(t)} = (MM^T)^t h^{(0)}$$

This provides the groundwork for us to demonstrate the convergence of the scores. In other words, we first wish to show that there exists $h^{(*)}$, such that:

$$\lim_{t\to\infty} \frac{h^t}{c^t} = h^{(*)},$$

where c is a constant. Using our above equations we have:

$$(MM^T)h^{(*)} = c \cdot h^{(*)}$$

We observe that MM^T is symmetric since $(MM^T)^T = (M^T)^T M^T = MM^T$. Therefore, there exist eigenvectors $z_1, \cdots, z_n$ of MM^T with corresponding eigenvalues $c_1, \cdots, c_n$ that form a basis for MM^T and such that for all $x \in \mathbb{R}^n$, there exist constants $p_1, \cdots, p_n \in \mathbb{R}$ such that

$$\begin{aligned} MM^T x &= MM^T(p_1 z_1 + \cdots + p_n z_n) \\ &= p_1 c_1 z_1 + \cdots + p_n c_n z_n \end{aligned}$$

If we assume without loss of generality that $|c_1| \geq \cdots \geq |c_n|$ and $|c_1| > |c_2|$, we can write $h^{(0)}$ as a linear combination of the eigenvectors:

$$h^{(0)} = q_1 z_1 + \cdots + q_n z_n$$

for some set of constants $q_1, \cdots, q_n \in \mathbb{R}$. Thus, we have:

$$h^{(t)} = (MM^T)^t h^{(0)}$$

$$h^{(t)} = c_1^t q_1 z_1 + \cdots + c_n^t q_n z_n$$

$$\frac{h^{(t)}}{c_1^t} = q_1 z_1 + \left(\frac{c_2}{c_1}\right)^t q_2 z_2 + \cdots + \left(\frac{c_n}{c_1}\right)^t q_n z_n.$$

From our assumption that $|c_1| > |c_i|$ for all $i \geq 2$, $\frac{h^{(t)}}{c_1^t}$ thus converges to $h^{(*)} = q_1 z_1$ as t approaches infinity.

Exercises: If we instead have $|c_1| = |c_2| = \cdots = |c_l|$ for $l \leq n$, do the set of hub scores still converge? If so, what do they converge to now? How could we apply our work above to show that authority scores also converge?